In the claims:

1. (canceled)

2. (canceled)

3. (currently amended): A method for compiling an application program, comprising: compiling said application program with a first set of compiler options in a manner that optimizes one application characteristic, subject to limits on another application characteristic to provide a first executable, compiling said application program with a second set of compiler options in a manner that optimizes said other application characteristic, subject to limits on said one application characteristic to provide a second executable, generating profile information from said first and second executables, generating and displaying sets of optimal solutions from said profile information wherein the sets have methods of compiling at the function level in a solution space generator, and automatically selecting and applying function level compiler options for said application program based upon selected optimal solutions so as to optimize said application program.

4. (previously amended): The method of claim 3, wherein said first set of options is for best computation time.

5. (previously amended): The method of claim 4, wherein said second set of options is for best code size.

6. (previously amended): The method of Claim 3, further comprising the step of analyzing said profile information against user supplied application characteristic constraints for selecting said compiler options by function.

7. (canceled)

8. (canceled)

9. (canceled)

10. (previously amended): A method for compiling an application program comprising the steps of: compiling said application with a different set of compiler options that optimizes one application characteristic, subject to limits on another application characteristic to provide two or more executables; generating profile information from said executables; applying said profile information to a solver; generating sets of useful solutions from said solver wherein the sets have methods for compiling each function; and selecting a solution for said application program using said useful solutions for subsequent compiling of said application.

11. (original): The method of Claim 10 wherein said selecting step includes displaying said useful solutions.

12. (previously amended): The method of Claim 10 wherein said generating sets of useful solutions step includes generating an efficient frontier curve of optimum solution points and displaying said curve of solution points.

13. (previously amended): The method of Claim 12 wherein said generating sets of useful solutions step includes a zoom window of a section of said curve of solution points.

14. (previously amended): The method of claim 10 wherein said generating sets of useful solutions step includes linear programming and heuristics to reduce the number of permutations of option sets per function.

15. (previously amended): The method of claim 10 wherein said generating solutions step generates possible solutions and filters out the possible solutions that are not better in at least one application characteristic.

16. (previously amended): The method of claim 15 wherein said generating solutions step includes a search tree wherein each candidate is applied to a node and compared to the solution at the node and if faster in time and smaller in size replacing that candidate at the node, if neither faster nor smaller in size discarding the candidate, if faster only processing down the tree in one direction and if smaller only processing down the tree in a different direction.

17. (previously amended): The method of claim 10 wherein the step of selecting a solution includes displaying solution points on said solution point curve illustrating the application characteristic tradeoff that can be made by compiling each function as prescribed by said solution.

18. (previously amended): The method of Claim 10 including means for overriding the solvers choice of compiler options for a particular function.

19. (previously amended): The method of Claim 17 including the step of compiling the application using chosen solution's set of function options.

20. (canceled)

21. (canceled)

22. (canceled)

23. (canceled)

24. (canceled)

25. (previously added): The method of claim 10 wherein the step of selecting a solution includes applying a user constraint to one application characteristic and automatically selecting the solution point that meets said constraint and optimizes another application constraint.

26. (canceled)

27. (cancelled)

28. (canceled)

29. (canceled)

30. (canceled)

31. (canceled)

32 (canceled)

33. (canceled)